UNITED STATES PATENT APPLICATION

For

DOCUMENT REGISTRATION

Inventors:

Ratinder Paul Singh Ahuja
Erik de la Iglesia
Rick Lowe
Matthew Howard

Prepared by:
Blakely, Sokoloff, Taylor & Zafman
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(408) 947-8200

Attorney's Docket No. 6897P007

"Express Mail" mailing label number: EV410138146US
Date of Deposit: March 30, 2004
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Vineta T. Tufono
(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

3-30-2004
(Date signed)

DOCUMENT REGISTRATION

PRIORITY AND RELATED APPLICATIONS

[0001] This patent application is related to, incorporates by reference, and claims the priority benefit of U.S. Provisional Application 60/528,631, entitled "DOCUMENT REGISTRATION", filed December 10, 2003.

FIELD OF THE INVENTION

[0002] The present invention relates to computer networks, and in particular, to registering documents in a computer network.

BACKGROUND

[0003] Computer networks and systems have become indispensable tools for modern business. Modern enterprises use such networks for communications and for storage. The information and data stored on the network of a business enterprise is often a highly valuable asset. Modern enterprises use numerous tools to keep outsiders, intruders, and unauthorized personnel from accessing valuable information stored on the network. These tools include firewalls, intrusion detection systems, and packet sniffer devices. However, once an intruder has gained access to sensitive content, there is no network device that can prevent the electronic transmission of the content from the network to outside the network. Similarly, there is no network device that can analyse the data leaving the network to monitor for policy violations, and make it possible to

track down information leeks. What is needed is a comprehensive system to capture, store, and analyse all data communicated using the enterprises network.

SUMMARY OF THE INVENTION

[0004] A document accessible over a network can be registered. A registered document, and the content contained therein, cannot be transmitted undetected over and off of the network. In one embodiment, the invention includes maintaining a plurality of stored signatures, each signature being associated with one of a plurality of registered documents, intercepting an object being transmitted over a network, calculating a set of signatures associated with the intercepted object, and comparing the set of signatures with the plurality of stored signatures. In one embodiment, the invention can further include detecting registered content from the registered document being contained in the intercepted object, if the comparison results in a match of at least one of the signatures in the set of signatures with one or more of the plurality of stored signatures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements and in which:

[0006] **Figure 1** is a block diagram illustrating a computer network connected to the Internet;

[0007] **Figure 2** is a block diagram illustrating one configuration of a capture system according to one embodiment of the present invention;

[0008] **Figure 3** is a block diagram illustrating the capture system according to one embodiment of the present invention;

[0009] **Figure 4** is a block diagram illustrating an object assembly module according to one embodiment of the present invention;

[0010] **Figure 5** is a block diagram illustrating an object store module according to one embodiment of the present invention;

[0011] **Figure 6** is a block diagram illustrating an example hardware architecture for a capture system according to one embodiment of the present invention;

[0012] **Figure 7** is a block diagram illustrating a document registration system according to one embodiment of the present invention;

[0013] **Figure 8** is a block diagram illustrating registration module according to one embodiment of the present invention; and

[0014] **Figure 9** is a flow diagram illustrating object capture processing according to one embodiment of the present invention.

DETAILED DESCRIPTION

[0015] Although the present system will be discussed with reference to various illustrated examples, these examples should not be read to limit the broader spirit and scope of the present invention. Some portions of the detailed description that follows are presented in terms of algorithms and symbolic representations of operations on data within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the computer science arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared and otherwise manipulated.

[0016] It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise, it will be appreciated that throughout the description of the present invention, use of terms such as "processing", "computing", "calculating", "determining", "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented

as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0017] As indicated above, one embodiment of the present invention is instantiated in computer software, that is, computer readable instructions, which, when executed by one or more computer processors/systems, instruct the processors/systems to perform the designated actions. Such computer software may be resident in one or more computer readable media, such as hard drives, CD-ROMs, DVD-ROMs, read-only memory, read-write memory and so on. Such software may be distributed on one or more of these media, or may be made available for download across one or more computer networks (e.g., the Internet). Regardless of the format, the computer programming, rendering and processing techniques discussed herein are simply examples of the types of programming, rendering and processing techniques that may be used to implement aspects of the present invention. These examples should in no way limit the present invention, which is best understood with reference to the claims that follow this description.

Networks

[0018] Figure 1 illustrates a simple prior art configuration of a local area network (LAN) 10 connected to the Internet 12. Connected to the LAN 102 are various components, such as servers 14, clients 16, and switch 18. There are numerous other known networking components and computing devices that can be connected to the LAN 10. The LAN 10 can be implemented using various wireline or wireless technologies,

such as Ethernet and 802.11b. The LAN 10 may be much more complex than the simplified diagram in Figure 1, and may be connected to other LANs as well.

[0019] In Figure 1, the LAN 10 is connected to the Internet 12 via a router 20. This router 20 can be used to implement a firewall, which are widely used to give users of the LAN 10 secure access to the Internet 12 as well as to separate a company's public Web server (can be one of the servers 14) from its internal network, i.e., LAN 10. In one embodiment, any data leaving the LAN 10 towards the Internet 12 must pass through the router 12. However, there the router 20 merely forwards packets to the Internet 12. The router 20 cannot capture, analyse, and searchably store the content contained in the forwarded packets.

[0020] One embodiment of the present invention is now illustrated with reference to Figure 2. Figure 2 shows the same simplified configuration of connecting the LAN 10 to the Internet 12 via the router 20. However, in Figure 2, the router 20 is also connected to a capture system 22. In one embodiment, the router 12 splits the outgoing data stream, and forwards one copy to the Internet 12 and the other copy to the capture system 22.

[0021] There are various other possible configurations. For example, the router 12 can also forward a copy of all incoming data to the capture system 22 as well. Furthermore, the capture system 22 can be configured sequentially in front of, or behind the router 20, however this makes the capture system 22 a critical component in connecting to the Internet 12. In systems where a router 12 is not used at all, the capture system can be interposed directly between the LAN 10 and the Internet 12. In one embodiment, the capture system 22 has a user interface accessible from a LAN-attached device, such as a client 16.

[0022] In one embodiment, the capture system 22 intercepts all data leaving the network. In other embodiments, the capture system can also intercept all data being communicated inside the network 10. In one embodiment, the capture system 22 reconstructs the documents leaving the network 10, and stores them in a searchable fashion. The capture system 22 can then be used to search and sort through all documents that have left the network 10. There are many reasons such documents may be of interest, including network security reasons, intellectual property concerns, corporate governance regulations, and other corporate policy concerns.

Capture System

[0023] One embodiment of the present invention is now described with reference to Figure 3. Figure 3 shows one embodiment of the capture system 22 in more detail. The capture system 22 is also sometimes referred to as a content analyzer, content or data analysis system, and other similar names. In one embodiment, the capture system 22 includes a network interface module 24 to receive the data from the network 10 or the router 20. In one embodiment, the network interface module 24 is implemented using one or more network interface cards (NIC), e.g., Ethernet cards. In one embodiment, the router 20 delivers all data leaving the network to the network interface module 24.

[0024] The captured raw data is then passed to a packet capture module 26. In one embodiment, the packet capture module 26 extracts data packets from the data stream received from the network interface module 24. In one embodiment, the packet capture module 26 reconstructs Ethernet packets from multiple sources to multiple destinations for the raw data stream.

[0025] In one embodiment, the packets are then provided the object assembly module 28. The object assembly module 28 reconstructs the objects being transmitted by the packets. For example, when a document is transmitted, e.g. as an email attachment, it is broken down into packets according to various data transfer protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP) and Ethernet. The object assembly module 28 can reconstruct the document from the captured packets.

[0026] One embodiment of the object assembly module 28 is now described in more detail with reference to Figure 4. When packets first enter the object assembly module, they are first provided to a reassembler 36. In one embodiment, the reassembler 36 groups – assembles – the packets into unique flows. For example, a flow can be defined as packets with identical Source IP and Destination IP addresses as well as identical TCP Source and Destination Ports. That is, the reassembler 36 can organize a packet stream by sender and recipient.

[0027] In one embodiment, the reassembler 36 begins a new flow upon the observation of a starting packet defined by the data transfer protocol. For a TCP/IP embodiment, the starting packet is generally referred to as the "SYN" packet. The flow can terminate upon observation of a finishing packet, e.g., a "Reset" or "FIN" packet in TCP/IP. If now finishing packet is observed by the reassembler 36 within some time constraint, it can terminate the flow via a timeout mechanism.. In an embodiment using the TPC protocol, a TCP flow contains an ordered sequence of packets that can be assembled into a contiguous data stream by the ressembler 36. Thus, in one embodiment, a flow is an ordered data stream of a single communication between a source and a destination.

[0028] The flown assembled by the reassember 36 can then be provided to a protocol demultiplexer (demux) 38. In one embodiment, the protocol demux 38 sorts assembled flows using the TCP Ports. This can include performing a speculative classification of the flow contents based on the association of well-known port numbers with specified protocols. For example, Web Hyper Text Transfer Protocol (HTTP) packets – i.e., Web traffic – are typically associated with port 80, File Transfer Protocol (FTP) packets with port 20, Kerberos authentication packets with port 88, and so on. Thus in one embodiment, the protocol demux 38 separates all the different protocols in one flow.

[0029] In one embodiment, a protocol classifier 40 also sorts the flows in addition to the protocol demux 38. In one embodiment, the protocol classifier 40 – operating either in parallel or in sequence with the protocol demux 38 – applies signature filters to the flows to attempt to identify the protocol based solely on the transported data. Furthermore, the protocol demux 38 can make a classification decision based on port number which is subsequently overridden by protocol classifier 40. For example, if an individual or program attempted to masquerade an illicit communication (such as file sharing) using an apparently benign port such as port 80 (commonly used for HTTP Web browsing), the protocol classifier 40 would use protocol signatures, i.e., the characteristic data sequences of defined protocols, to verify the speculative classification performed by protocol demux 38.

[0030] In one embodiment, the object assembly module 28 outputs each flow organized by protocol, which represent the underlying objects. Referring again to Figure 3, these objects can then be handed over to the object classification module 30

(sometimes also referred to as the "content classifier") for classification based on content. A classified flow may still contain multiple content objects depending on the protocol used. For example, protocols such as HTTP (Internet Web Surfing) may contain over 100 objects of any number of content types in a single flow. To deconstruct the flow, each object contained in the flow is individually extracted, and decoded, if necessary, by the object classification module 30.

[0031] The object classification module 30 uses the inherent properties and signatures of various documents to determine the content type of each object. For example, a Word document has a signature that is distinct from a PowerPoint document, or an Email document. The object classification module 30 can extract out each individual object and sort them out by such content types. Such classification renders the present invention immune from cases where a malicious user has altered a file extension or other property in an attempt to avoid detection of illicit activity.

[0032] In one embodiment, the object classification module 30 determines whether each object should be stored or discarded. In one embodiment, this determination is based on a various capture rules. For example, a capture rule can indicate that Web Traffic should be discarded. Another capture rule can indicate that all PowerPoint documents should be stored, except for ones originating from the CEO's IP address. Such capture rules can be implemented as regular expressions, or by other similar means.

[0033] In one embodiment, the capture rules are authored by users of the capture system 22. The capture system 22 is made accessible to any network-connected machine through the network interface module 24 and user interface 34. In one embodiment, the

user interface 34 is a graphical user interface providing the user with friendly access to the various features of the capture system 22. For example, the user interface 34 can provide a capture rule authoring tool that allows users to write and implement any capture rule desired, which are then applied by the object classification module 30 when determining whether each object should be stored. The user interface 34 can also provide pre-configured capture rules that the user can select from along with an explanation of the operation of such standard included capture rules. In one embodiment, the default capture rule implemented by the object classification module 30 captures all objects leaving the network 10.

[0034] If the capture of an object is mandated by the capture rules, the object classification module 30 can also determine where in the object store module 32 the captured object should be stored. With reference to Figure 5, in one embodiment, the objects are stored in a content store 44 memory block. Within the content store 44 are files 46 divided up by content type. Thus, for example, if the object classification module determines that an object is a Word document that should be stored, it can store it in the file 46 reserved for Word documents. In one embodiment, the object store module 32 is integrally included in the capture system 22. In other embodiments, the object store module can be external – entirely or in part – using, for example, some network storage technique such as network attached storage (NAS) and storage area network (SAN).

[0035] In one embodiment, the content store is a canonical storage location, simply a place to deposit the captured objects. The indexing of the objects stored in the content store 44 is accomplished using a tag database 42. In one embodiment, the tag database 42 is a database data structure in which each record is a "tag" that indexes an

object in the content store 44, and contains relevant information about the stored object. An example of a tag record in the tag database 42 that indexes an object stored in the content store 44 is set forth in Table 1:

Table 1

Field Name	Definition
MAC Address	Ethernet controller MAC address unique to each capture system
Source IP	Source Ethernet IP Address of object
Destination IP	Destination Ethernet IP Address of object
Source Port	Source TCP/IP Port number of object
Destination Port	Destination TCP/IP Port number of the object
Protocol	IP Protocol that carried the object
Instance	Canonical count identifying object within a protocol capable of carrying multiple data within a single TCP/IP connection
Content	Content type of the object
Encoding	Encoding used by the protocol carrying object
Size	Size of object
Timestamp	Time that the object was captured
Owner	User requesting the capture of object (rule author)
Configuration	Capture rule directing the capture of object
Signature	Hash signature of object
Tag Signature	Hash signature of all preceding tag fields

[0036] There are various other possible tag fields, and some embodiments can omit numerous tag fields listed in Table 1. In other embodiments, the tag database 42 need not be implemented as a database; other data structures can be used. The mapping of tags to objects can, in one embodiment, be obtained by using unique combinations of tag fields to construct an object's name. For example, one such possible combination is an ordered list of the Source IP, Destination IP, Source Port, Destination Port, Instance and Timestamp. Many other such combinations including both shorter and longer names are possible. In another embodiment, the tag can contain a pointer to the storage location where the indexed object is stored.

[0037] Referring again to Figure 3, in one embodiment, the objects and tags stored in the object store module 32 can be interactively queried by a user via the user interface 34. In one embodiment the user interface can interact with a web server (not shown) to provide the user with Web-based access to the capture system 22. The objects in the content store module 32 can thus be searched for specific textual or graphical content using exact matches, patterns, keywords, and various other advanced attributes.

[0038] For example, the user interface 34 can provide a query-authoring tool (not shown) to enable users to create complex searches of the object store module 32. These search queries can be provided to a data mining engine (not shown) that parses the queries, scans the tag database 42, and retrieves the found object from the content store 44. Then, these objects that matched the specific search criteria in the user-authored query can be counted and displayed to the user by the user interface 34.

[0039] Searches can also be scheduled to occur at specific times or at regular intervals, that is, the user interface 34 can provide access to a scheduler (not shown) that can periodically execute specific queries. Reports containing the results of these searches can be made available to the user at a later time, mailed to the administrator electronically, or used to generate an alarm in the form of an e-mail message, page, syslog or other notification format.

[0040] In several embodiments, the capture system 22 has been described above as a stand-alone device. However, the capture system of the present invention can be implemented on any appliance capable of capturing and analysing data from a network. For example, the capture system 22 described above could be implemented on one or

more of the servers 14 or clients 16 shown in Figure 1. The capture system 22 can interface with the network 10 in any number of ways, including wirelessly.

[0041] In one embodiment, the capture system 22 is an appliance constructed using commonly available computing equipment and storage systems capable of supporting the software requirements. In one embodiment, illustrated by Figure 6, the hardware consists of a capture entity 46, a processing complex 48 made up of one or more processors, a memory complex 50 made up of one or more memory elements such as RAM and ROM, and storage complex 52, such as a set of one or more hard drives or other digital or analog storage means. In another embodiment, the storage complex 52 is external to the capture system 22, as explained above. In one embodiment, the memory complex stored software consisting of an operating system for the capture system device 22, a capture program, and classification program, a database, a filestore, an analysis engine and a graphical user interface.

Document Registration

[0042] The capture system 22 described above can also be used to implement a document registration scheme. In one embodiment, the a user can register a document with the system 22, which can then alert the user if all or part of the content in the registered document is leaving the network. Thus, one embodiment of the present invention aims to prevent un-authorized documents of various formats (e.g., Microsoft Word, Excel, PowerPoint, source code of any kind, text) from leaving an enterprise. There are great benefits to any enterprise that can keep its intellectual property, or other critical, confidential, or otherwise private and proprietary content from being mishandled.

[0043] In one embodiment of the present invention, sensitive documents are registered with the capture system 22, although data registration can be implemented using a separate device in other embodiments. One embodiment of implementing registration capability in the capture system 22 is now described with reference to Figure 7. For descriptive purposes, the capture system 22 is renamed the capture/registration system 22 in Figure 7, and is also sometimes referred to as the registration system 22 in the description herein. The capture/registration system 22 has components similar or identical to the capture system 22 shown in Figure 3, including the network interface module 24, the object store module 32, the user interface 34, and the packet capture 26, object assembly 28, and object classification 30 modules, which are grouped together as object capture modules 31 in Figure 7.

[0044] In one embodiment, the capture/registration system 22 also includes a registration module 54 interacting with a signature database 56 to facilitate a registration scheme. In one embodiment, the user can register a document via the user interface 34. There are numerous ways to register documents. For example, a document can be electrically mailed (e-mailed), or uploaded to the registration system 22. The registration system 22 can also periodically scan a file server (registration server) for documents to be registered. The registration process can be integrated with the enterprise's document management systems. Document registration can also be automated and transparent based on registration rules, such as "register all documents," or "register all documents by specific author or IP address," and so on.

[0045] After being received, in one embodiment, a document to be registered is passed to the registration module 54. The registration module 54 calculates a signature of

the document, or a set of signatures. The set of signatures associated with the document can be calculated in various ways. For example, the signatures can be made up of hashes over various portions of the document, such as selected or all pages, paragraphs, tables and sentences. Other possible signatures include, but are not limited to, hashes over embedded content, indices, headers or footers, formatting information or font utilization. The signatures can also include computations and meta-data other than hash digests, such as Word Relative Frequency Methods (RFM) – Statistical, Karp-Rabin Greedy-String-Tiling-Transposition, vector space models, and diagrammatic structure analysis.

[0046] The set of signatures is then stored in the signature database 56. The signature database 56 need not be implemented as a database; the signatures can be maintained using any appropriate data structure. In one embodiment, the signature database 56 is part of the storage complex 52 in Figure 6.

[0047] In one embodiment, the registered document is also stored as an object in the object store module 32. In one embodiment, the document is only stored in the content store 44 with no associated tag, since many tag fields do not apply to registered documents. In one embodiment, one file of files 46 is a "Registered Documents" file.

[0048] In one embodiment, the document received from the user is now registered. As set forth above, in one embodiment, the object capture modules 31 continue to extract objects leaving the network, and store various objects based on capture rules. In one embodiment, all extracted objects – whether subject to a capture rule or not – are also passed to the registration module for a determination whether each object is, or includes part of, a registered document.

[0049] In one embodiment, the registration module 54 calculates the set of signatures of an object received from the object capture modules 31 in the same manner as of a document received from the user interface 34 to be registered. This set of signatures is then compared against all signatures in the signature database 56. In other embodiment, parts of the signature database can be excluded from this search to save time.

[0050] In one embodiment, an unauthorized transmission is detected if any one or more signatures in the set of signatures of an extracted object matches one or more signature in the signature database 56 associated with a registered document. Other detection tolerances can be configured for different embodiment, e.g., at least two signatures must match. Also, special rules can be implemented that make the transmission authorized, e.g., if the source address is authorized to transmit any documents off the network.

[0051] One embodiment of the registration module 54 is now described with reference to Figure 8. As discussed above, a document to be registered 68 arrives via the user interface 34. The registration engine 58 generates signatures 60 for the document 68 and forwards the document 68 to the content store 44 and the signatures 60 to the signature database 54. The signatures 60 are associated with the document, e.g., by including a pointer to the document 68, or to some attribute from which the document 68 can be identified.

[0052] A captured object 70 arrives via the object capture modules 31. The registration engine calculates the signatures 62 of the captured object, and forwards them to the search engine 64. The search engine 64 queries the signature database 54 to

compare the signatures 62 to the signatures stored in the signature database 54. Assuming for the purposes of illustration, that the captured object 70 is a Word document that contains a pasted paragraph from registered PowerPoint document 68, at least one signature of signatures 62 will match a signature of signatures 60. Such an event can be referred to as the detection of an unauthorized transfer, a registered content transfer, or other similarly descriptive terms.

[0053] In one embodiment, when a registered content transfer is detected, the transmission can be halted with or without warning to the sender. In one embodiment, in the event of a detected registered content transfer, the search engine 64 activates the notification module 66, which sends an alert 72 to the user via the user interface 34. In one embodiment, the notification module 66 sends different alerts – including different user options – based on the user preference associated with the registration, and the capabilities of the registration system 22.

[0054] In one embodiment, the alert 72 can simply indicate that the registered content, i.e., the captured object 70, has been transferred off the network. In addition, the alert 72 can provide information regarding the transfer, such as source IP, destination IP, any other information contained in the tag of the captured object, or some other derived information, such as the name of the person who transferred the document off the network. The alert 72 can be provided to one or more users via e-mail, instant message (IM), page, or any other notification method. In one embodiment, the alert 72 is only sent to the entity or user who requested registration of the document 68.

[0055] In another embodiment, the delivery of the captured object 70 is halted – the transfer is not completed – unless the user who registered the document 68 consents.

In such an embodiment, the alert 72 can contain all information described above, and in addition, contain a selection mechanism, such as one or two buttons – to allow the user to indicate whether the transfer of the captured object 70 may be completed. If the user elects to allow the transfer, for example because he is aware that someone is emailing a part of a registered document 68 (e.g., a boss asking his secretary to send an email), the transfer is executed and the object 70 is allowed to leave the network.

[0056] If the user disallows the transfer, the captured object 70 is not allowed off the network, and delivery is permanently halted. In one embodiment, halting delivery can be accomplished by implementing an intercept technique by having the registration system 22 proxy the connection between the network and the outside. In other embodiments, delivery can be halted using a black hole technique – discarding the packets without notice if the transfer is disallowed – or a poison technique – inserting additional packets onto the network to cause the sender's connection to fail.

[0057] Figure 9 provides a flow chart to further illustrate object capture/intercept processing according to one embodiment of the present invention. All blocks of Figure 9 have already been discussed herein. The example object capture processing shown in Figure 9 assumes that various documents have already been registered with the registration system 22. The process shown in Figure 9 can be repeated for all objects captured by the system 22.

[0058] Thus, a capture system and a document/content registration system have been described. In the forgoing description, various specific values were given names, such as "objects," and various specific modules, such as the "registration module" and "signature database" have been described. However, these names are merely to describe

and illustrate various aspects of the present invention, and in no way limit the scope of the present invention. Furthermore, various modules, such as the search engine 64 and the notification module 66 in Figure 8, can be implemented as software or hardware modules, or without dividing their functionalities into modules at all. The present invention is not limited to any modular architecture either in software or in hardware, whether described above or not.